

Operating Systems

Lecture 04: Familiarizing yourself with linux

Familiarizing yourself with linux

Shell Commands:

- **GUI:** The Linux Graphical User Interface (GUI) is a collection of icons, windows, and other screen graphical elements that help users interact with the operating system. The desktop menu provides access to the GUI applications available on the Linux desktop. There are different GUI implementations such as K Desktop Environment (KDE) and GNU Object Model Environment (GNOME).

Familiarizing yourself with linux

File system Hierarchy Standard (FHS)

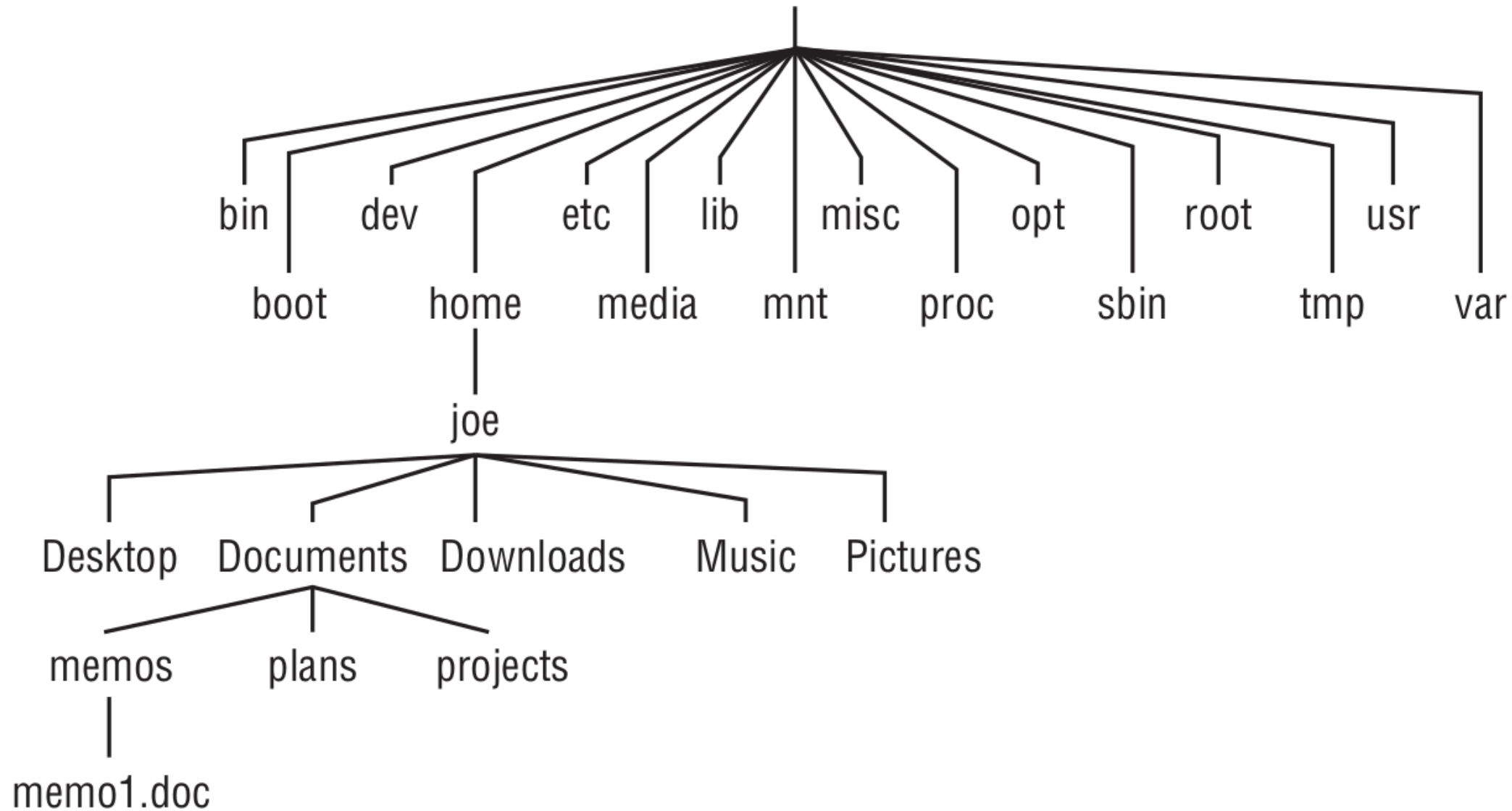
- The Linux file system is the structure in which all the information on your computer is stored.
- In fact, one of the defining properties of the UNIX systems on which Linux is based is that nearly everything you need to identify on your system (data, commands, symbolic links, devices, and directories) is represented by items in the file systems.

Familiarizing yourself with linux

File system Hierarchy Standard (FHS)

- The Linux file system is the structure in which all the information on your computer is stored.
- In fact, one of the defining properties of the UNIX systems on which Linux is based is that nearly everything you need to identify on your system (data, commands, symbolic links, devices, and directories) is represented by items in the file systems.

The Linux filesystem is organized as a hierarchy of directories.



Familiarizing yourself with linux

Some of these Linux directories may interest you:

- **/bin:** Contains common Linux user commands, such as ls, sort, date, and chmod.
- **/boot:** Has the bootable Linux kernel and boot loader configuration files (GRUB).
- **/dev:** Contains files representing access points to devices on your systems. These include terminal devices (tty*), floppy disks (fd*), hard disks (hd* or sd*), RAM (ram*), and CD-ROM (cd*). Users can access these devices directly through these device files; however, applications often hide the actual device names from end users.

Familiarizing yourself with linux

- **/etc:** Contains administrative configuration files. Most of these files are plaintext files that can be edited with any text editor if the user has proper permission.
- **/home:** Contains directories assigned to each regular user with a login account. (The root user is an exception, using /root as his or her home directory.)
- **/media:** Provides a standard location for automounting devices (removable media in particular). If the medium has a volume name, that name is typically used as the mount point. For example, a USB drive with a volume name of myusb would be mounted on /media/myusb.

Familiarizing yourself with linux

- **/lib:** Contains shared libraries needed by applications in /bin and /sbin to boot the system.
- **/mnt:** A common mount point for many devices before it was supplanted by the standard /media directory. Some bootable Linux systems still use this directory to mount hard disk partitions and remote filesystems. Many people still use this directory to temporarily mount local or remote filesystems that are not mounted permanently.
- **/misc:** A directory sometimes used to automount filesystems upon request.
- **/opt:** Directory structure available to store add-on application software.

Familiarizing yourself with linux

- **/proc:** Contains information about system resources.
- **/root:** Represents the root user's home directory. The home directory for root does not reside beneath /home for security reasons.
- **/sbin:** Contains administrative commands and daemon processes.
- **/tmp:** Contains temporary files used by applications.
- **/usr:** Contains user documentation, games, graphical files (X11), libraries (lib), and a variety of other commands and files that are not needed during the boot process. The /usr directory is meant for files that don't change after installation (in theory, /usr could be mounted read-only).

Familiarizing yourself with linux

- **/var:** Contains directories of data used by various applications. In particular, this is where you would place files that you share as an FTP server (/var/ftp) or a web server (/var/www). It also contains all system log files (/var/log) and spool files in /var/spool (such as mail, cups, and news). The /var directory contains directories and files that are meant to change often. On server computers, it is common to create the /var directory as a separate filesystem, using a filesystem type that can be easily expanded.

Familiarizing yourself with linux

Shell Commands:

- **The Command Line Interface (CLI)** is a text-based interface for the operating system, where a user typically enters commands at the command prompt to instruct the computer to perform a specific task.
- **A command line interpreter**, or command line shell, is a program that implements the commands entered in the text interface. The command line interpreter analyzes the input text provided by the user, interprets the text in the concept given, and then provides the output.

Familiarizing yourself with linux

Shell Commands:

- **GUI:** The Linux Graphical User Interface (GUI) is a collection of icons, windows, and other screen graphical elements that help users interact with the operating system. The desktop menu provides access to the GUI applications available on the Linux desktop. There are different GUI implementations such as K Desktop Environment (KDE) and GNU Object Model Environment (GNOME).

Familiarizing yourself with linux

Shell Commands:

- **A shell:** is a component that interacts directly with users. It also functions as the command interpreter for the Linux system. The shell accepts user commands and ensures that the kernel carries them out. The shell also contains an interpretive programming language.

The various shells available in Linux are described in the following table.

Shell	Description
Bash	This is the default Linux shell. It provides the flexibility of the C shell in a Bourne shell-type environment. Use the command <code>bash</code> to open the Bash shell.
Bourne	This is the original Unix shell developed by Steve Bourne at Bell Labs and is available on all Linux systems. Use the command <code>sh</code> to open the Bourne shell.
C shell	This was developed by Bill Joy at Berkeley and was designed to support C language development environments. It was also designed for more interactive use, providing several ways to reduce the amount of typing needed to complete a job. Use the command <code>csh</code> to open the C shell.
Korn	This shell is a combination of the C and Bourne shells. It uses the features of the C shell but the syntax of the Bourne shell. Use the command <code>ksh</code> to open the Korn shell.

Familiarizing yourself with linux

Shell Commands:

- **Virtual Terminals:** A terminal or console is a computer interface for text entry and display, where information is displayed as an array of preselected characters. Linux supports six virtual terminals in the CLI mode, which provide a text terminal with a login prompt to the shell. You can choose among these six terminals by using the key combination of Ctrl+Alt+F1 → F6. You can be logged in to multiple virtual terminals at the same time.

Familiarizing yourself with linux

Shell Commands:

- The generic format for a shell command is **(command -option argument)**.
- After typing your command, the shell responds by performing a specific action that is associated with that command.
- Linux is case sensitive, so you must enter commands in the required case.
- **Argument:** also called command line argument, is usually a file name or directory name that indicates the files on which the command will operate. It is used as an input by some commands in Linux. Arguments can be files, directories, commands or even a command switch.

For example, ls {file name}, ls {directory name}, and ls -l.

Familiarizing yourself with linux

Shell Commands:

- **Command History**

- Sometimes, commands can become quite long. You can access previously entered commands that are stored in the History file by using the Up Arrow and the Down Arrow keys.
- The history file is stored in `~/.bash_history`
- To list all command you have written, type **history** in terminal.

Familiarizing yourself with linux

Shell Commands:

- **Invoking Commands Outside a Path**

- There are two ways of invoking a command located outside a path.
- You can specify the path in which the command is located and then invoke the command. For example, assume that a command is located in the `{user-defined directory}` directory. To invoke this command, you need to enter `{user-defined directory}/{command name}`.
- You can also navigate to the directory that contains the command and then invoke it. For example, assume that a command is located in the `{user-defined directory}` directory. You need to change to that directory with the `cd {user-defined directory}` command and then enter `./{command name}`.

Familiarizing yourself with linux

Shell Commands:

- **The Tab-Completion Feature**

- Some commands have long names containing version number information, weird spellings, or capitalizations. This can make it difficult to correctly enter the commands on the first try.
- Enter the first few characters of the command and then press Tab. If there is only one match, the rest of the command name is displayed
- If there is similar starting names for the command, then you should press Tab twice to choose the required command.

Familiarizing yourself with linux

Shell Commands:

- **Piping Commands**

- To send or redirect the results of one command to another command. Pipes are used to combine Linux tools on a single command line, enabling you to use the output of one command as the input to another. The pipe symbol is a vertical bar (|), which you type between two commands. For example, `ls | more` enables you to look at a large directory listing one screen at a time.

- **Issuing More Than One Command**

- To issue more than one command before pressing Enter. Place a semicolon (;) between the commands and they will be issued one after the other.

Familiarizing yourself with linux

Shell Commands:

- **The exec Command**

- If you enter a command, it runs as a child process to Bash, which is the parent process. If you enter `exec {command}`, the `exec` command will kill the parent process, the `bash` process, and `{command}` starts to run as the parent process. For example, when a user has a limit applied on the number of process, the user can use the `exec` command to run an additional process by killing the parent process. Once the `exec {command}` is executed, you will be automatically logged out because the `bash` process has been terminated.

Familiarizing yourself with linux

Shell Commands:

- **The date Command**

- The syntax of the date command is `date +[format]`, where format is the string of characters that are used to display the different fields of the output.
- Example : `date +%d:%m-%Y`

- **The cal Command**

- The cal command displays the calendar for any month or year
- The syntax of the cal command is `cal {month} {year}`, Ex: `cal 4 2000`

- **The uptime Command**

- The uptime command displays the time from when a system started running.
- The output of the uptime command gives information about the current time, how long the system is running, and how many users are currently logged in.
- The last field of the uptime command output displays the system's load averages for the last 1 minute, 5 minutes, and 15 minutes. This information can be used to check whether the system is busy.

Familiarizing yourself with linux

Shell Commands:

- **The who Command**

- The who command is used to determine the details of users currently logged in to a system.
- The output of the who command includes the user name, the name of the system from which the user is connected, and the time since the user is connected.

- **The whoami Command**

- The whoami command is used to display the user name with which you are currently logged in to the system.

- **The hostname Command**

- The hostname command is used to display the hostname of the system you are currently logged in to.

Familiarizing yourself with linux

Shell Commands:

- **The w Command**

- The w command is primarily used to display the details of users who are currently logged in to a system and their transactions.
- The first line of the output displays the status of the system.
- The second line of the output displays a table with the first column listing the users logged in to the system and the last column indicating the current activities of the users.
- The remaining columns of the table show different attributes associated with the users.
- The w command retrieves information from /var/run/utmp and /proc

Familiarizing yourself with linux

Shell Commands:

- **The last Command**

- The last command displays the history of user log in and log out, It also has options that enable you to filter users who have logged in through a specific terminal.
- For example, last 1 will display the details of users who logged in using the first terminal.
- The “last” command retrieves information from /var/log/wtmp file

- **The echo Command**

- The echo command is used to display a line of text on the terminal. It is useful for programmers writing shell scripts because it can be used to display additional information. The text that needs to be displayed should be inserted after the echo command. You can also use the echo command to display the value stored in a variable by specifying the variable name after the echo command.

Familiarizing yourself with linux

Shell Commands:

- **The sleep Command**

- The command `sleep {time}` hangs up the prompt for the number of seconds specified by the value of the variable `time`. Ex: `sleep 5`

- **The which Command**

- The `which` command is used to verify whether a user has the right to execute a command. It displays the complete path of the command by searching the directories assigned to the `PATH` variable.
- For example, on entering “`which cat`”, the following output is displayed: `/bin/cat`.

Familiarizing yourself with linux

Get Help Using Linux

- **Linux Documentation**

- Linux documentation is the material that provides information on various Linux commands and blocks of code.
- **System documentation**
 - Is the term given to the collection of documents that list the system requirements;
 - Its functioning capabilities, limitations, design specifications; the internal workings of the system; and the steps for maintaining the system.

- **Manual Pages**

- The syntax of the man command is `man {topic}`.

- **The apropos Command**

- The syntax of the apropos command is `apropos {keyword}`.

Familiarizing yourself with linux

Get Help Using Linux

- **Other Built-In Help Options**

- **Whatis**

- Displays a short description of the command along with the man pages/sections matching the exact command.
- The syntax of this command is `what is {command}`

- **Info**

- Displays info pages containing additional or recent information about a command.
- The syntax of this command is `info {command}`.

- **command -help**

- Displays a quick summary of the usage of a command and a list of arguments that can be used.
- This feature can be used with most commands in Linux.
- The syntax of this command is `command -options`.

- **The /usr/share/doc Directory**

- The /usr/share/doc directory contains documents installed on the system, describing in detail certain aspects of configuring or using Linux.

Familiarizing yourself with linux

Get Help Using Linux

- **Other Built-In Help Options**

- **HOWTOs**

- HOWTOs are comprehensive documents, much like FAQs, but generally not in question-and-answer format. However, many HOWTOs contain a FAQ section at the End.
- HOWTOs can be found on most systems and can also be found on the web at <http://linuxdocs.org/HOWTOs/HOWTO-INDEX/howtos.html> and other sites

Familiarizing yourself with linux

Get Help Using Linux

- **Other Built-In Help Options**

- **LUGs**

- A good source of information for Linux users and developers is Linux User Groups, or LUGs. These can be virtual (based on the web) or there may be a group of people who meet in your neighborhood.

Familiarizing yourself with linux

Get Help Using Linux

- **Online Help**
 - <http://linuxdocs.org/>
 - <https://askubuntu.com/>
 - <https://unix.stackexchange.com/>
 - <https://stackoverflow.com/>
 - <https://itsfoss.com/>
 - <http://www.yourownlinux.com/>
- **Alternative windows and macOS programs in linux**
 - <https://alternativeto.net/platform/linux/>

Thanks For Attention